National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

Self-Adaptive Discovery Mechanisms for Improved Performance in Fault-Tolerant Networks

Kevin Mills, Doug Montgomery, Scott Rose,
Stephen Quirolgico, Kevin Bowers, and Chris Dabrowski

DARPA FTN PI Meeting
January 27, 2003

Scalable Software for
Hostile & Volatile Environments

Presentation Outline

- One-Page Review of Project Objective and Plan
- One-Page Refresher on Service Discovery Protocols
- Analysis of Jini Leasing Performance
- Self-Adaptive Leasing for Jini
  - Two Algorithms: Simple Adaptive Leasing and Inverted Leasing
  - Performance characteristics (*obtained via simulation*)
- Leasing with Multiple Lookup Services
- Summary of Other Accomplishments Since July 2002
- Plan for Next Six Months
- Conclusions

Project Objective

Research, design, evaluate, and implement self-adaptive mechanisms to improve performance of service discovery protocols for use in fault-tolerant networks.

Project Plan – Three Phases

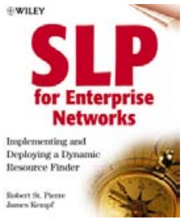
- Phase I – characterize performance of selected service discovery protocols (Universal Plug-and-Play – UPnP – and Jini) as specified and implemented
  - develop simulation models for each protocol
  - establish performance benchmarks based on default or recommended parameter values and on required or most likely implementation of behaviors
- Phase II – design, simulate, and evaluate self-adaptive algorithms to improve performance of discovery protocols regarding selected mechanisms
  - devise algorithms to adjust control parameters and behavior in each protocol
  - simulate performance of each algorithm against benchmark performance
  - select most promising algorithms for further development
- Phase III – implement and validate the most promising algorithms in publicly available reference software

Dynamic Discovery Protocols in Essence

Dynamic discovery protocols enable *network elements*:

(1) to *discover* each other without prior arrangement,
(2) to *express* opportunities for collaboration,
(3) to *compose* themselves into larger collections that cooperate to meet an application need, and
(4) to *detect and adapt to changes* in network topology.

Selected First-Generation Dynamic Discovery Protocols

JINI 3-Party Design	UPnP FORUM 2-Party Design	 WILEY SLP for Enterprise Networks Implementing and Deploying a Dynamic Resource Finder Robert St. Pierre James Kempf — Adaptive 2/3-Party Design
The Salutation Consortium Vertically Integrated 3-Party Design	HAVi Network-Dependent 3-Party Design	Bluetooth Network-Dependent 2-Party Design

A Brief History of Leases in Distributed Systems

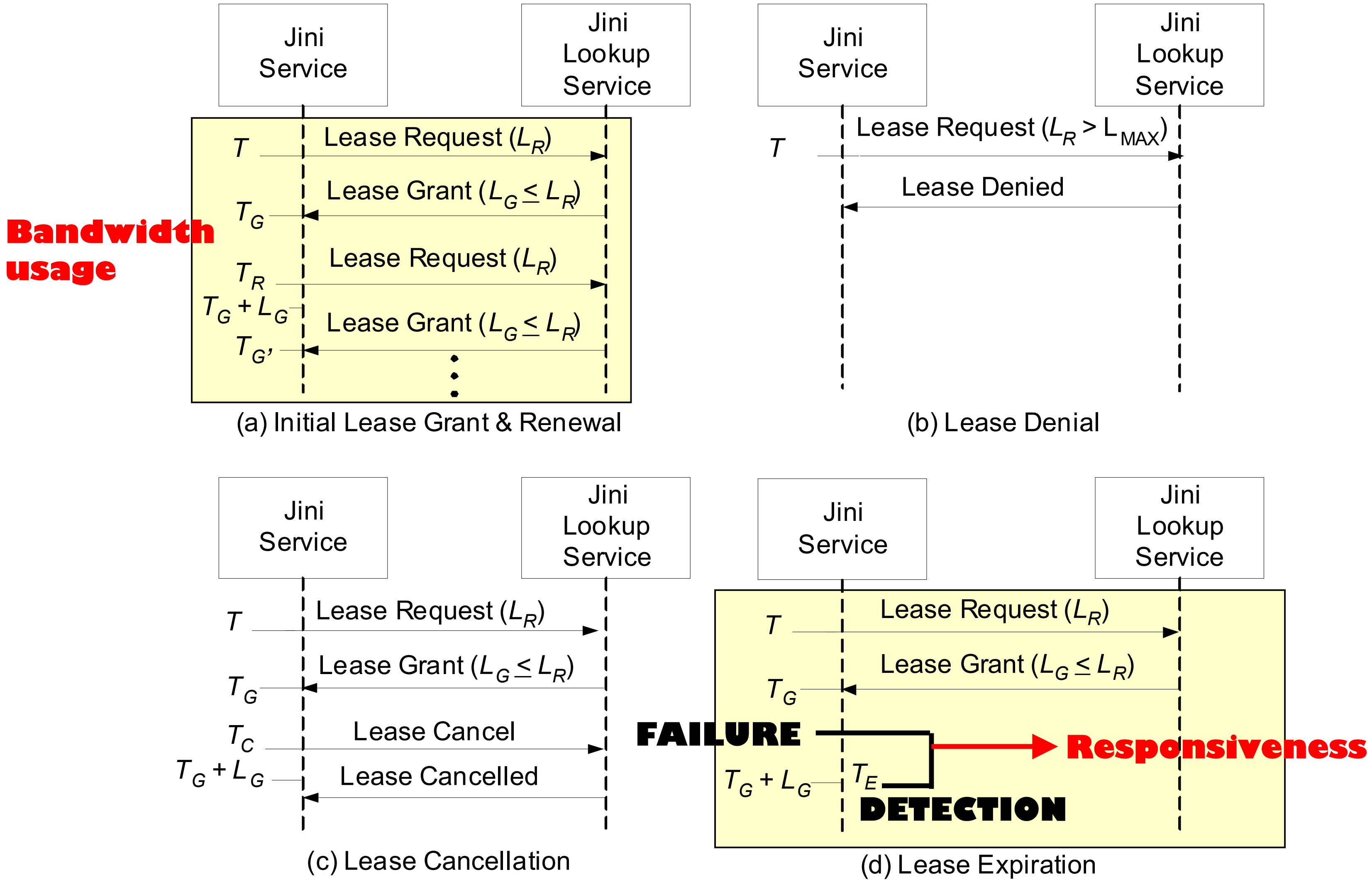
- Originally proposed by Gray and Cheriton for consistency maintenance in distributed file caches [Gray and Cheriton. "Leases: an efficient fault-tolerant mechanism for distributed file cache consistency", *ACM SIGOPS Operating Systems Review*, November 1989.]

- Now widely used in distributed systems
  - Mobile Networking
    - Cao, "On improving the performance of cache invalidation in mobile environments", *Mobile Networks and Applications,* August 2002.
    - Perkins and Luo, "Using DHCP with computers that move0", *Wireless Networks*, March 1995.
    - Zheng, Ge , Hou , and Thuel, "A case for mobility support with temporary home agents", ACM *SIGMOBILE Mobile Computing and Communications Review*, January 2002.
  - Distributed File Systems
    - Grönvall, Westerlund, and Pink. "The design of a multicast-based distributed file system", *Proceedings of the third symposium on Operating systems design and implementation*, February 1999
    - Mann, Birrell, Hisgen , Jerian , and Swart. "A coherent distributed file cache with directory write-behind", ACM *Transactions on Computer Systems* (TOCS), May 1994.
    - Muthitacharoen, Chen, and Mazières. "A low-bandwidth network file system," ACM *SIGOPS Operating Systems Review*, October 2001
    - Thekkath, Mann, and Lee. "Frangipani: a scalable distributed file system", ACM *SIGOPS Operating Systems Review*, October 1997.

A Brief History of Leases in Distributed Systems (cont.)

- Shared Memory
  - Harris and Sarkar. "Lightweight object-oriented shared variables for distributed applications on the Internet", ACM *SIGPLAN Notices*, October 1998.
  - Gharachorloo, Gupta, and Hennessy. " Performance evaluation of memory consistency models for shared-memory multiprocessors", ACM *SIGARCH Computer Architecture News*, April 1991.
- Web Systems
  - Ninan, Kulkarni, Shenoy, Ramamritham, and Tewari. "Performance: Cooperative leases: scalable consistency maintenance in content distribution networks", *Proceedings of the eleventh international conference on World Wide Web*, May 2002.
  - Jacobsen and Günther. "Middleware for software leasing over the Internet", *Proceedings of the first ACM conference on Electronic commerce*, November 1999.
  - Shih and Shim. "A service management framework for M-commerce applications", *Mobile Networks and Applications*, June 2002.
- Service Discovery Systems
  - Friday, Davies, and Catterall. "Supporting service discovery, querying and interaction in ubiquitous computing environments", *Second ACM international workshop on Data engineering for wireless and mobile access*, May 2001.
  - Hodes, Czerwinski, Zhao, Joseph, and Katz. "An architecture for secure wide-area service discovery", *Wireless Networks*, March 2002.
  - Universal Plug and Play Device Architecture, Version 1.0, 08 Jun 2000 10:41 AM. © 1999-2000 Microsoft Corporation. All rights reserved.
  - Waldo. "The Jini architecture for network-centric computing", *Communications of the ACM*, July 1999.

Selected Jini Leasing Sequences



Analysis of Jini Leasing Performance

Let N = number of leaseholders, S_R = size of lease request message, and S_G = size of lease grant message

Bandwidth Consumption (B)

$$B = (N / L_G) \cdot (S_R + S_G)$$

Responsiveness (R)

$$R = L_G / 2$$

Given requirements for B and R, what lease period should be granted to each leaseholder and how many leaseholders can be supported?

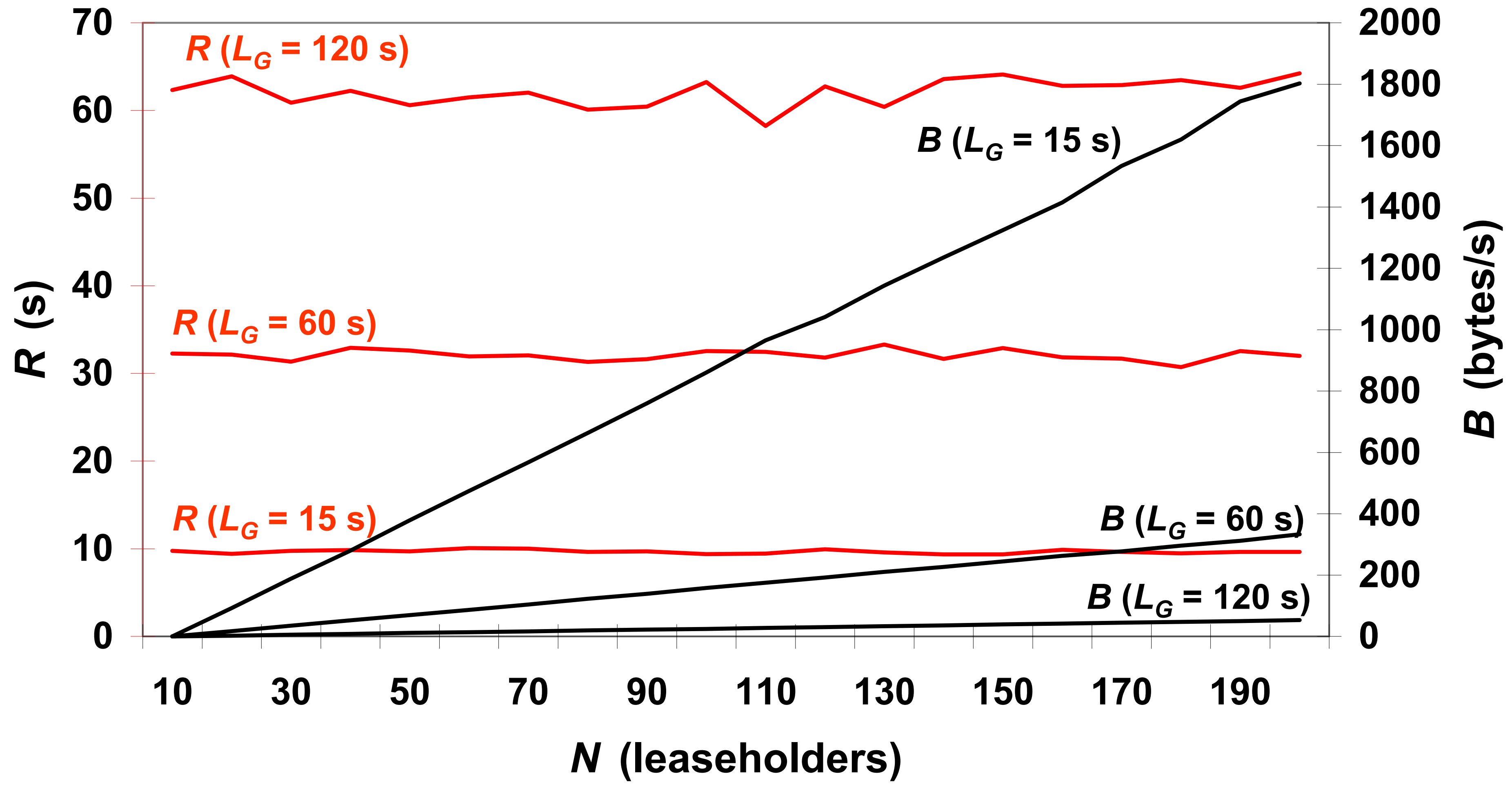
$L_G = 2R$ Re-writing responsiveness equation yields a value for lease period to grant

$N_{MAX} = (B \cdot L_G)/(S_R + S_G)$ Transforming bandwidth equation indicates maximum system capacity

What decisions must the lease grantor make to guarantee R and B?

1. Deny lease requests that would consume excessive bandwidth (*i.e.*, when $L_R < L_G$)
2. Grant lease periods no greater than L_G to ensure desired responsiveness
3. Deny lease requests when the number of leaseholders would exceed capacity (*i.e.*, when $N = N_{MAX}$)

Simulation Results: Responsiveness and Bandwidth Usage vs. Network Size for Various L_G Values



A Simple Adaptive Leasing Mechanism

Goal: limit bandwidth usage to *B* and guarantee a minimum responsiveness (R_{MIN}), while achieving the best possible responsiveness $R > R_{MIN}$ when $N < N_{MAX}$

Preliminary Analysis

$L_{MAX} = 2R_{MIN}$	Minimum Responsiveness determines maximum granted lease period
$G = B/(S_R + S_G)$	Available bandwidth determines maximum lease renewals per second (*G*)
$L_{MIN} = 1/G$	Assuming minimum system size of 1, *G* determines minimum granted lease period
$L_{MIN} = 2R_{MAX}$	However, 1/*G* might place too great a load on the leaseholder, so instead choose a maximum responsiveness and let that determine the minimum granted lease period

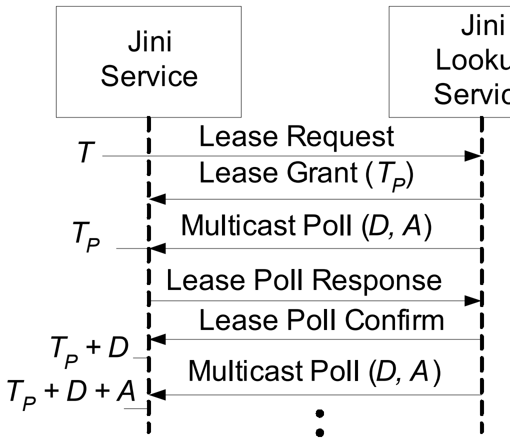
$L_{MIN} \leq L_G \leq L_{MAX}$ Vary the granted lease period within this range, using the following algorithm

Adaptive Algorithm for Varying L_G

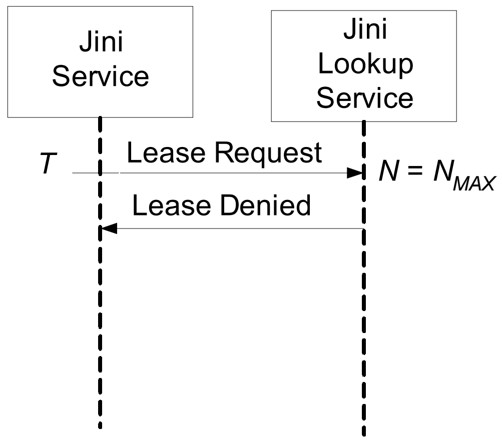
```
set $L_G = N / G$;
if $L_G > L_{MAX}$
    then deny the lease;
    elseif $L_G < L_{MIN}$
        then set $L_G = L_{MIN}$;
    endif
endif
```

An Inverted Leasing Mechanism

Main Idea: lookup service periodically polls leaseholders on a multicast channel – adapting the polling interval to accommodate variations in the number of leaseholders



(a) Initial Lease Grant & Renewal

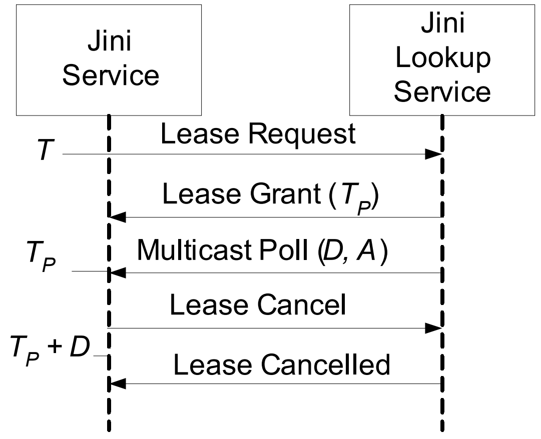


(b) Lease Denial

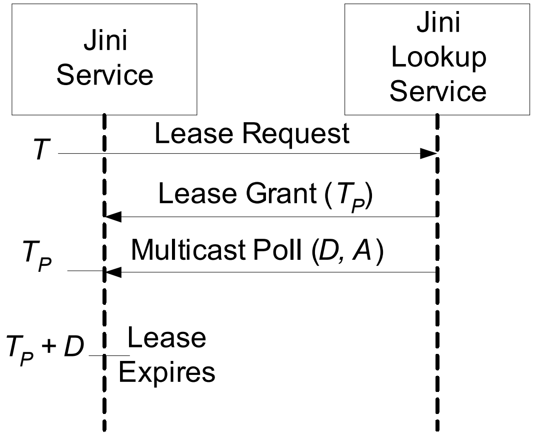
Polling Interval

Each poll contains the interval (D) over which the lookup service listens for responses and an additional time (A) within which the next poll will be sent, and

$$L_{MIN} \leq D + A \leq L_{MAX}$$



(c) Lease Cancellation



(d) Lease Expiration

Additional Constraint

Grant leases only up to $N \leq N_{MAX}$, where $N_{MAX} = L_{MAX} * G$

Adaptive Algorithm for Inverted Leasing Mechanism

Adaptive Algorithm for Varying *D* (and *A*) and Selecting T_{POLL}

```
set $D$ = Max ($N / G, L_{MIN}$);
set $A = 0.2\ D$
if $D + A > L_{MAX}$
        then set $A = 0$;
endif
set $T_{POLL}$ = time + $D$;
```

Since $N \leq N_{MAX}$, $D \leq L_{MAX}$

Preliminary Analysis

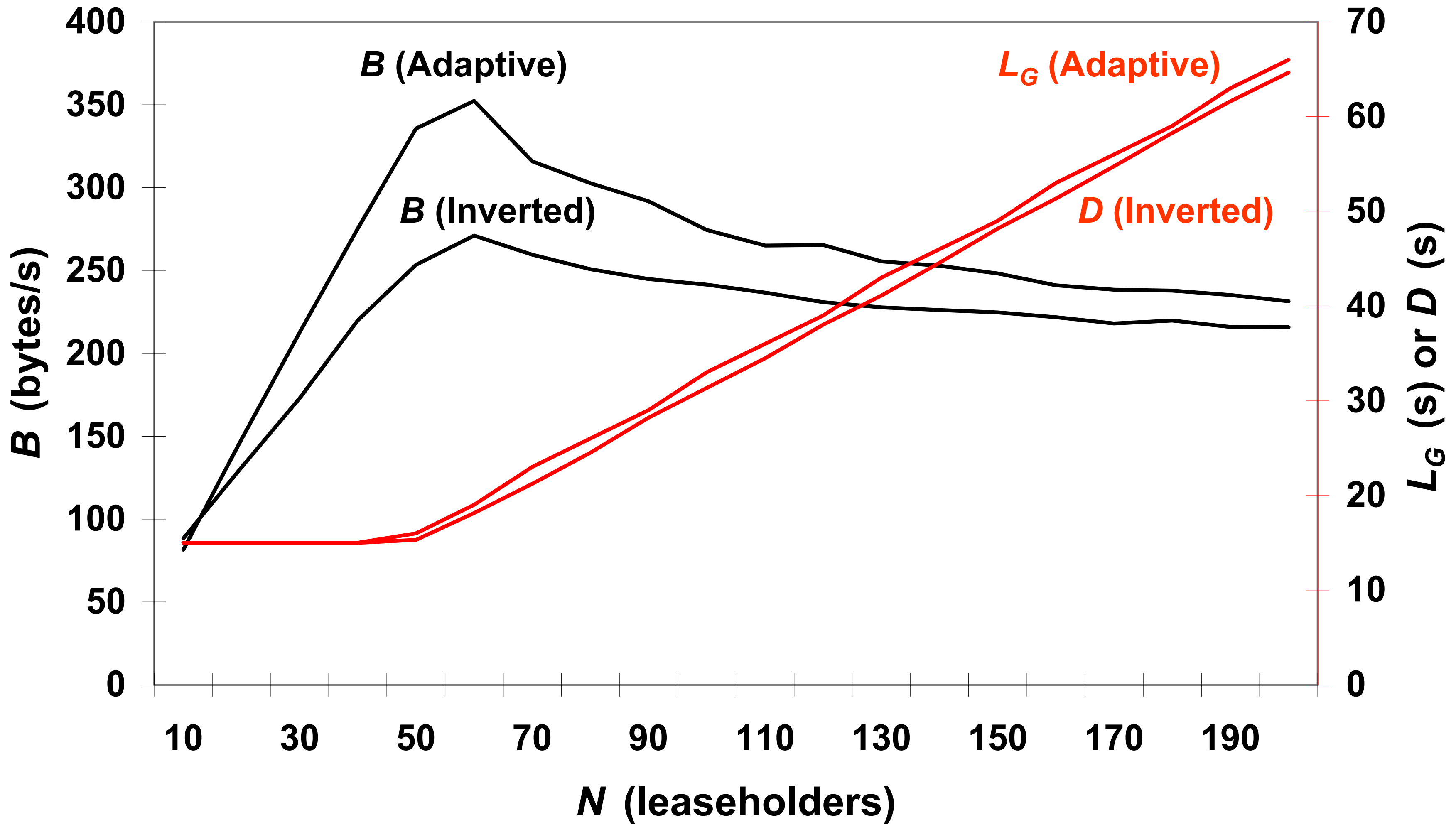
$$B = S_P + ((N/P)\cdot(S_{PR} + S_{RC}))$$ where *P* is the polling interval ($D \leq P \leq D + A \leq L_{MAX}$), S_P is poll size, and S_{PR} and S_{RC} are size of poll response and confirm

Assuming half of failures occur before poll and half after: $R = 1/2\cdot(D/2) + 1/2\cdot(3D/2) = D$

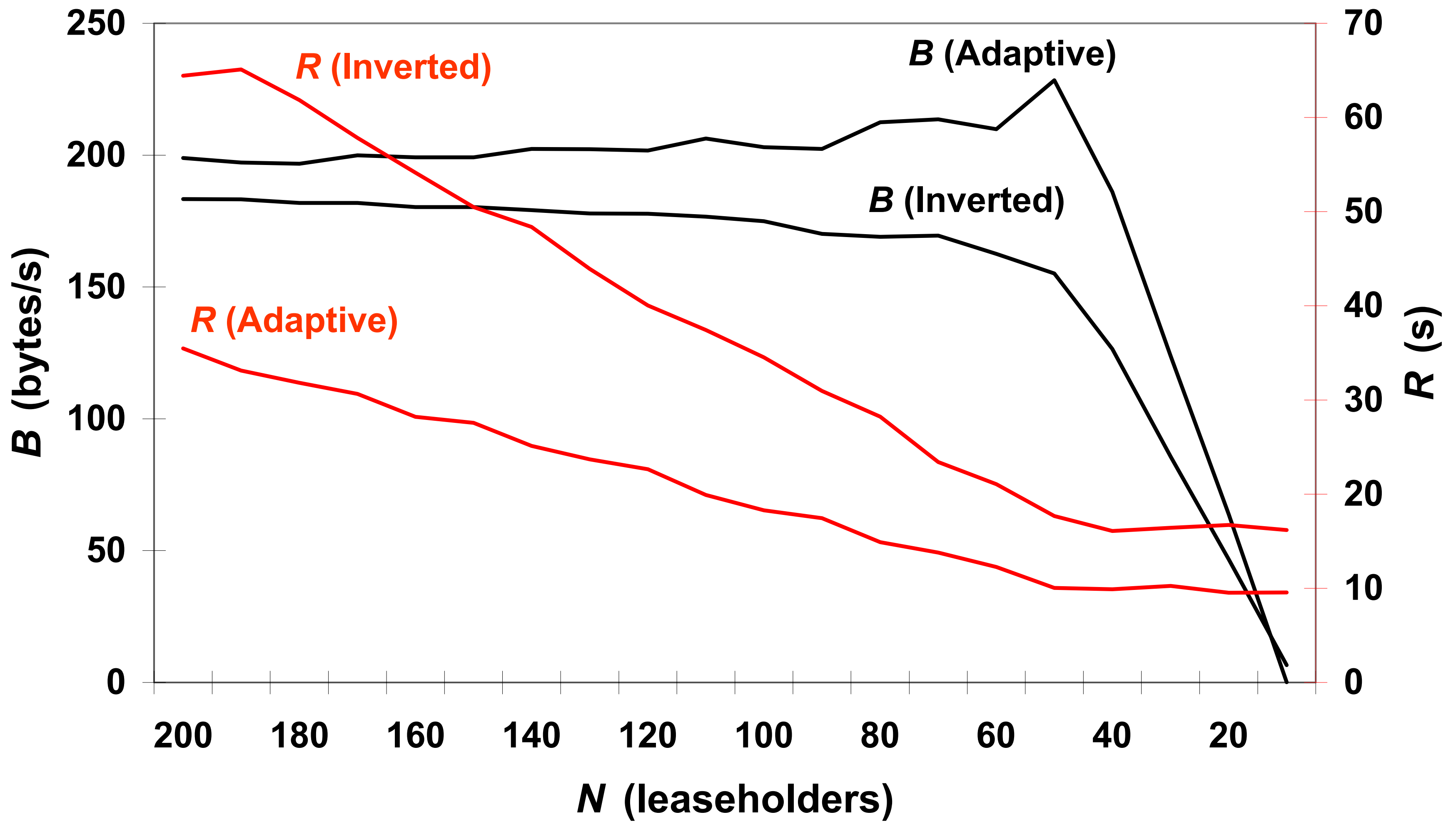
So inverted leasing will be only ½ as responsive as simple adaptive leasing (recall $R = L_G / 2$)

From this result we can also infer that $R_{MAX} = L_{MIN}$ and $R_{MIN} = L_{MAX}$

Bandwidth Usage and Control Variable Value vs. Increasing Network Size for Adaptive and Inverted Leasing Algorithms



Bandwidth Usage and Responsiveness vs. Decreasing Network Size for Adaptive and Inverted Leasing Algorithms



Adaptive Leasing with Multiple Lookup Services

Main Goal: Given a domain-wide limit for leasing resources, expressed either in terms of bandwidth (B_D) or renewals per second (G_D), the main goal is to allocate a fair share of the resources to each lease grantor within the domain.

- Let N_D represent the number of lookup services within a domain.
- Assume each Jini lookup service is configured with a network-wide resource budget for leasing (either B_D or G_D)
- Each lookup service can compute its share of available resources (either B_D / N_D or G_D / N_D).

But we need a means to increase and decrease the allocation of resources with changes in N_D

- Jini facilitates monitoring N_D by requiring each lookup service to announce itself periodically (every 120 s recommended) on a designated multicast channel.
- Each lookup service can increment N_D when a new lookup service is heard and can decrement N_D when an expected announcement is missed.

Adaptive Algorithm for Varying N_D

As N_D varies, each lookup service can continuously adjust its share of the available domain-wide leasing resources.

Other Accomplishments Since July 2002

- Produced three papers
  - "Understanding Self-healing in Service Discovery Systems", C. Dabrowski and K. Mills, *Proceedings of ACM SigSoft Workshop on Self-healing Systems*, November 2002, Charleston, SC, pp. 15-20.
  - “Adaptive Jitter Control for UPnP M-Search", K. Mills and C. Dabrowski, accepted by IEEE International Conference on Communications, 2003.
  - “Self-Adaptive Leasing for Jini”, K. Bowers, K. Mills, and S. Rose, accepted by IEEE Pervasive Computing (PerCom) 2003 conference.
- Completed characterization of UPnP and Jini behavior in a tactical (multiple sensor-actuator) application during node failure
- Completed scalable (up to 500 nodes) discrete-event simulation model of the Service Location Protocol (SLP)

Plan for the Next Six Months

- Implement our simple self-adaptive leasing in the publicly available Jini reference code distributed by Sun Microsystems – and demonstrate it at the 2003 DARPA Information Survivability Conference and Exposition (DISCEX III) in April (we will show this demonstration again at the summer 2003 FTN PI meeting)
- Characterize the behavior of the service location protocol (SLP) under hostile and volatile conditions – expect a journal paper in 2003 characterizing the performance of Jini, UPnP, and SLP in response to power failure, communication failure, message loss, and node failure
- Formalize a generic model of service-discovery architectures, including structure, behavior, and properties – expect a journal paper in 2003
- Develop an analytical model of the consistency maintenance behavior of Jini and UPnP during communication failure – expect a journal paper
- Investigate self-adaptive mechanisms for inclusion in SLP
- Continue interactions with the Sun Microsystems Jini team; with Microsoft, Intel and the UPnP Forum; and with the IETF SLP group

Conclusions

- Emerging industry discovery protocols exhibit performance characteristics that vary based on parameter settings, network size, and resource availability
- Tuning such dynamic systems cannot rely on manual configuration methods
- We illustrated one case – Jini leasing – where values for granted lease periods interact with system size to determine performance and resource usage
- We proposed two self-adaptive algorithms for Jini leasing, and we investigated relative performance of the algorithms
- We explained how the simple adaptive leasing algorithm could be used in a Jini system with multiple lookup services
- We believe that our simple adaptive leasing algorithm can also be used for UPnP event subscriptions and for SLP service registrations (with some adjustments).
- We have shared our findings with Sun, Microsoft, Intel, the UPnP Forum, and the IETF SLP group